IN THE CLAIMS

Please amend the claims as follows. Added text is underlined and deleted text is either struck through or shown in double enclosing brackets. Applicants aver that no new matter has been added.

1. (Currently Amended) A ~~computer implemented method for automatic software tuning, the~~ method comprising:

calculating ~~at least one~~ a first threshold value for ~~at least one~~ a first parameter and a second threshold value for a second parameter, the first parameter and the second parameter influencing the performance of a software application with regards to a specific task, the first threshold value separating a first value range of the first parameter into two intervals of a first dimension and the second threshold value separating a second value range of the second parameter into at least two intervals of a second dimension;

comparing the ~~at least one~~ first threshold value to ~~at least one~~ a corresponding current value of the first parameter and the second threshold value to a corresponding current value of the second parameter; and

selecting, using one or more processors, an algorithm from a plurality of algorithms for performing the task in accordance with the result of the comparing step, the selected algorithm assigned to an intersection of the interval of the first dimension that includes the corresponding current parameter value of the first dimension and the interval of the second dimension that includes the corresponding current parameter value of the second dimension.

2. (Currently Amended) The method of claim 1, further comprising:

measuring the performance of the selected algorithm;

checking whether the selected algorithm delivers the better performance within the plurality of algorithms ~~measured performance complies with the at least one threshold value~~; and

recalculating at least the first ~~at least one~~ threshold value if a further algorithm of the plurality of algorithms performs better in the intersection including the current parameter values of the first dimension and the second dimension, the recalculation performed so that the further algorithm gets automatically selected in the intersection defined by the at least one recalculated threshold value. ~~in case of non-compliance~~.

3.-6. (Canceled)

7. (Currently Amended) The method claim [[3]] 1, wherein each threshold value corresponds to a break-even point where two neighbouring algorithms have the same performance with respect to the corresponding dimension.

8.-9. (Canceled)

10. (Currently Amended) ~~A computer program product for dynamically selecting a data retriever implementation for retrieving data from a data storage system in response to a Boolean expression, the computer program product~~ The method of claim 1, further comprising:

~~a result counter to determine~~ determining a number of hits in response to ~~the~~ a Boolean expression;

~~a threshold evaluator to compare~~ comparing the number of hits with the [[a]] first threshold value of [[a]]the first dimension and ~~to compare~~ comparing the complexity of the Boolean expression with ~~a further~~ a second threshold value of [[a]]the second dimension;

~~a first data retriever to retrieve~~ retrieving the data using a first data retriever in case the number of hits is below the first threshold value of the first dimension and the complexity of the Boolean expression is above the ~~further~~ second threshold value of the second dimension;

~~a second data retriever to retrieve~~ retrieving the data using a second data retriever in case the number of hits is above the first threshold value of the first dimension and the complexity of the Boolean expression is above the ~~further~~ second threshold value of the second dimension;

~~a third data retriever to retrieve~~ retrieving the data using a third data retriever in case the number of hits is below the first threshold value of the first dimension and the complexity of the Boolean expression is below the ~~further~~ second threshold value of the second dimension; and

~~a forth data retriever to retrieve~~ retrieving the data using a fourth data retriever in case the number of hits is above the first threshold value of the first dimension and the complexity of the Boolean expression is below the ~~further~~ second threshold value of the second dimension.

11. (Currently Amended) The ~~computer program product~~ method of claim 10, further comprising:

~~a retrieval time measuring component to measure~~ measuring the time that is consumed by a selected data retriever for various numbers of hits; and

~~a threshold calculator to~~ dynamically determining ~~determine~~ the first threshold value and the ~~further~~ second threshold value on the basis of [[the]] results of the ~~retrieval time~~ measuring ~~component and to feed back the determined threshold values into the threshold evaluator~~.

12. (Currently Amended) The ~~computer program product~~ method of claim 11, wherein the ~~first data retriever is implemented by~~ retrieving of the data in case the number of hits is below the first threshold value of the first dimension and the complexity of the Boolean expression is above the second threshold value of the second dimension is performed using a general data retrieval algorithm using result flag instances.

13. (Currently Amended) The ~~computer program product~~ method of claim 11, wherein the ~~second data retriever is implemented by~~ retrieving of the data in case the number of hits is above the first threshold value of the first dimension and the complexity of the Boolean expression is above the second threshold value of the second dimension is performed using a general data retrieval algorithm using bit maps.

14. (Currently Amended) The ~~computer program product~~ method of claim 11, wherein the ~~third data retriever is implemented by~~ retrieving of the data in case the number of hits is below the first threshold value of the first dimension and the complexity of the Boolean expression is below the second threshold value of the second dimension is performed using a lean AND data retrieval algorithm using result flag instances.

15. (Currently Amended) The ~~computer program product~~ method of claim 11, wherein the ~~forth data retriever is implemented by~~ retrieving of the data in case the number of hits is above the first threshold value of the first dimension and the complexity of the Boolean expression is below the second threshold value of the second dimension is performed using a lean AND data retrieval algorithm using bit maps.

16. (Canceled)

17. (Currently Amended) A ~~computer system for running a software application, the~~ system comprising:

a memory to store variables for ~~storing at least one~~ a first threshold value for ~~at least one~~ a first parameter and at least a second threshold value for at least a second parameter, the first parameter and the second parameter influencing the performance of the software application with regards to a specific task, the first threshold value separating a first value range of the first parameter into two intervals of a first dimension and the second threshold value separating a second value range of the second parameter into two intervals of a second dimension, the first parameter and the second parameter values having initial values being set by running test cases for a plurality of algorithms for performing the specific task; and

a threshold evaluator, having one or more processors, ~~for comparing~~ to compare the ~~at least one~~ first threshold value to ~~at least one~~ a corresponding current value of the first parameter and the second threshold value to a corresponding current value of the second parameter, the interval of the first dimension and the interval of the second dimension including the corresponding current parameter value of the first dimension and the interval of the second dimension including the corresponding current parameter value of the second dimension define an intersection, the intersection being used by ~~allowing~~ the software application to select an algorithm assigned to the intersection from [[a]] the plurality of algorithms for performing the specific task in accordance with the result of comparison.

18. (Currently Amended) The ~~computer~~ system of claim 17, further comprising:

a threshold calculator [[for recalculating]] to recalculate [[the]] at least one of the threshold values ~~in case the actual performance of the selected algorithm is non-compliant with the at least one threshold value,~~ if a further algorithm of the plurality of algorithms performs better in the intersection including the current parameter values of the first and the second dimension, wherein the recalculation is performed so that the further algorithm is automatically selected in the intersection defined by the at least one recalculated threshold value.

19. -22. (Canceled)

23. (Currently Amended) The ~~computer~~ system of claim 19, wherein each threshold value corresponds to a break-even point where two neighbouring algorithms have the same performance with respect to the corresponding dimension.

24. (New) A non-transitory machine-readable medium having instructions embodied thereon that when executed by one or more processors, cause the one or more processors to perform a method, the method comprising:

calculating a first threshold value for a first parameter and a second threshold value for a second parameter, the first parameter and the second parameter influencing the performance of a software application with regards to a specific task, the first threshold value separating a first value range of the first parameter into two intervals of a first dimension and the second threshold value separating a second value range of the second parameter into at least two intervals of a second dimension;

comparing the first threshold value to a corresponding current value of the first parameter and the second threshold value to a corresponding current value of the second parameter; and

selecting, using one or more processors, an algorithm from a plurality of algorithms for performing the task in accordance with the result of the comparing step, the selected algorithm assigned to an intersection of the interval of the first dimension that includes the corresponding current parameter value of the first dimension and the interval of the second dimension that includes the corresponding current parameter value of the second dimension.